**Java Shortcuts:**

1.

Deque<int[]> stack = new ArrayDeque<int[]>(temperatures.length);

stack.push(new int[] { i, temperatures[i] });

2. List<Integer> integerlist = Arrays.asList(1,2);

4. list.toArray(new Object[0 or size]);

5.PriorityQueue<Map.Entry<Integer, Integer>> maxHeap =

new PriorityQueue<>((a,b)->(b.getValue()-a.getValue()));

6. Boundary check with Math.min

private int findNeighbors(int i, int j, int[][] board) {

int n = 0;

for (int x = Math.max(i - 1, 0); x <= Math.min(i + 1, board.length - 1) ; x++)

{

for (int y = Math.max(j - 1, 0); y <= Math.min(j + 1, board[0].length - 1); y++) {

if (!(i == x && j == y))

n = n + board[x][y];

}

}

return n;

}

7. for(int i=0;i<nums.length;i++)

{

if(i==0|| nums[i] != nums[i-1])

// print i

}

8.

HashMap<String, Set<String>> map = new HashMap<>();

Set<String> val1 = map.getOrDefault(w1, new HashSet<>());

val1.add(w2);

map.put(w1, val1);